SCALABLE, RELIABLE SESSION INITIATION PROTOCOL SIGNALING ROUTING NODE

AN APPLICATION FOR

UNITED STATES LETTERS PATENT

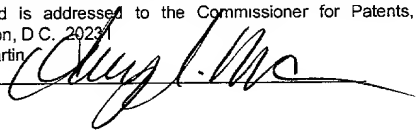
Express Mail Label No. EV 896 569 695 US

By

Richard Henry Sternagle

Raleigh, North Carolina

"Express Mail" mailing number EF227428751US
Date of Deposit 5-30-01
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C F R. 1.10 on the date indicated above and is addressed to the Commissioner for Patents, Washington, D C. 20231
Amy J Martin



Description

SCALABLE, RELIABLE SESSION INITIATION PROTOCOL (SIP) SIGNALING ROUTING NODE

Technical Field

The present invention relates to the session initiation protocol. More particularly, the present invention relates to a scalable, reliable session initiation protocol signaling routing node.

Background Art

The session initiation protocol or "SIP" is an application-layer control protocol for creating, modifying, and terminating sessions between communicating parties. The sessions include Internet multimedia conferences, Internet telephone calls, and multimedia distribution. Members in a session can communicate via unicast, multicast, or a mesh of unicast communications.

The SIP protocol is described in Handley et. al., *SIP: Session Initiation Protocol*, Internet Engineering Task Force (IETF) Request for Comments (RFC) 2543, March, 1999, the disclosure of which is incorporated herein by reference in its entirety. A related protocol used to describe sessions between communicating parties is the session description protocol. The session description protocol is described in Handley and Jacobsen, *SDP: Session Description Protocol*, IETF RFC 2327, April 1998, the disclosure of which is incorporated herein by reference in its entirety.

The SIP protocol defines several types of entities involved in establishing sessions between calling and called parties. These entities include: proxy servers, redirect servers, user agent clients, and user agent servers. A proxy server is an intermediary program that acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are serviced internally or by passing them on, possibly after translation to other servers. A proxy interprets, and, if necessary, rewrites a request message before forwarding the request. An example of a request in the SIP protocol is an INVITE message used to invite the recipient to participate in a session.

A user agent server is an application that contacts a user when a SIP request is received and that returns a response on behalf of the user. A user agent client is an application that initiates a SIP request. In a SIP network, user agent client and server software may execute on an end user device, such as a SIP telephone, to initiate and accept calls on behalf of the user.

A redirect server accepts a SIP request, maps the address into zero or more new addresses and returns these addresses to a client. Unlike a proxy server, a redirect server does not initiate its own SIP requests. Unlike a user agent server, a redirect server does not accept calls.

According to the SIP protocol, objects are identified by SIP URLs. A SIP URL may be of the form user@host. The user part may be a user name or a telephone number. A called party may move between a number of different systems or locations over time. These locations may be dynamically registered with a SIP server. A location server may also use one or more other protocols, such as finger, rwhois, LDAP, multicast protocols, or other operating-system-

Exhibit 1002 - Page 5 of 9

dependent mechanisms to locate an end system where the called party might be reachable.

Figure 1 is a message flow diagram recreated from the above-referenced SIP protocol specification that illustrates the function of the SIP proxy server in establishing a typical SIP call. In Figure 1, a user with a SIP URL of cz@cs.tu-berlin.de located at a first computer **100** in the domain cs.berlin.de is attempting to establish a call with another user with a SIP URL of henning@cs.col. In order to initiate the call, in step 1, the SIP user agent software resident on computer **100** sends an INVITE message to a SIP proxy server **102**. The INVITE message includes the SIP URL of the called party, i.e., henning@cs.col. Since SIP proxy server **102** may not know the actual location of the user henning@cs.col, in step 2, SIP proxy server **102** queries a location server **104** to determine where to send the INVITE message. In step 3, location server **104** responds with the current location of the user "henning". In the illustrated example, the location is specified as hgs@lab.

In step 4, SIP proxy server **102** forwards the INVITE message to computer **106** at which the user "henning" is located. SIP user agent software resident on computer **106** responds to the INVITE message with a 200 OKAY message indicating acceptance to the invitation to the session. In step 7, SIP proxy server **102** forwards the 200 OKAY message to computer **100**. In step 8, computer **100** forwards an acknowledgement message to SIP proxy server **102**. In step 9, SIP proxy server **102** forwards the acknowledgement message to computer **106**. Once the acknowledgement is received by computer **106**, a multi-media session is established between the two users.

Page 3 of 9 Ex. 1009

One potential problem that is not addressed in the SIP protocol specification is how to reliably and efficiently provide location information to SIP servers, such as SIP proxy servers and SIP redirect servers. Conventional SIP servers utilize a centralized database, as illustrated in Figure 1, to obtain SIP location information. This solution is undesirable because using a centralized server to provide the location information causes a performance bottleneck at the location server. That is, as the number of subscribers and location queries increase, the location server can become overwhelmed with location requests. As a result, location requests may be delayed or even dropped by the location server.

Another problem that is not addressed by the SIP protocol specification is how to provide reliability and scalability in SIP protocol servers, such as proxy servers and redirect servers. As the number of SIP users increases, the demands on SIP protocol servers will also increase. If a SIP protocol server fails, users may be left without SIP signaling service. The SIP protocol specification does not address methods of increasing scalability or reliability of SIP protocol servers. The SIP protocol specification merely discusses the functional requirements of these servers, in general.

Thus, there exists a long felt need for a scalable, reliable SIP signaling router that avoids at least some of the difficulties not addressed by the SIP protocol specification or by conventional SIP signaling routers.

Disclosure of the Invention

According to one aspect, the present invention includes a scalable, reliable, SIP signaling router. The SIP signaling router includes a plurality of cluster nodes for performing at least one SIP protocol function, such as SIP

proxy services or SIP redirect services. Each of the cluster nodes stores a local database including SIP location information. A location server is coupled to each of the cluster nodes for maintaining a database of SIP location information. The location server automatically replicates the database of SIP location information to each of the cluster nodes in real time in response to receiving updates to the SIP location information. Because the location server replicates a copy of its database to each of the cluster nodes, the cluster nodes can respond to SIP queries faster than conventional SIP proxy servers that are required to access an external location server to obtain SIP location information.

According to another aspect, the present invention includes a method for monitoring the operational status of cluster nodes performing SIP protocol functions, load sharing between the cluster nodes based on the operational status, and rerouting messages in the event of failure of one of the cluster nodes. In order to determine the operational status, an Ethernet switch periodically sends health check and ping messages to each of the plurality of cluster nodes. Operational status may be determined based on the response time for the ping and health check messages. The Ethernet switch may also maintain a connection tuple table that includes entries storing connection information for connections serviced by each of the cluster nodes. The load balancing may be performed based on the response time to the ping and health check messages and the number of connections in progress with each of the cluster nodes, as evidenced by the connection tuple table for each node.

In order to maintain reliable connectivity between the cluster nodes and external networks, a standby Ethernet switch is provided in addition to the active Ethernet switch. The active Ethernet switch replicates its connection tuple table

Express Mail No. EV 328 676 505 US

to the standby Ethernet switch using a spanning tree algorithm. Each of the cluster nodes includes a connection to the active Ethernet switch and a connection to the standby Ethernet switch. In the event of failure of the active Ethernet switch, operation automatically switches to the standby Ethernet switch.

Accordingly, it is an object of the present invention to provide a scalable, reliable SIP signaling router.

It is another object of the present invention to provide a SIP signaling router in which a location server replicates its database of SIP location information to SIP cluster nodes that perform SIP protocol functions.

It is yet another object of the present invention to provide operational status monitoring, load sharing, and reliable network connection for cluster nodes performing SIP protocol functions.

DEDZNOSOP

Brief Description of the Drawings

Preferred embodiments of the invention will now be explained with reference to the accompanying drawings, of which:

Figure 1 is a message flow diagram illustrating the functionality of a conventional SIP proxy server in establishing a SIP session;

Figure 2 is a block diagram of a scalable, reliable SIP signaling router according to an embodiment of the present invention;

Figure 3 is a flow diagram illustrating exemplary steps for replicating a SIP location database from a location server to a plurality of SIP protocol servers according to an embodiment of the present invention;

Figure 4 is a flow chart illustrating exemplary steps for incremental loading of the SIP location database maintained by standby location server **206**;

Figure 5 is a flow chart illustrating exemplary steps that may be performed in incremental loading of a cluster node database according to an embodiment of the present invention;

Figure 6 is a flow chart illustrating exemplary steps for continuous cluster node database reloading according to an embodiment of the present invention;

Figure 7 is a flow chart illustrating exemplary steps for incremental cluster node database loading according to an embodiment of the present invention;

Figure 8 is a block diagram illustrating a method for monitoring the operational status of cluster nodes providing SIP protocol services according to an embodiment of the present invention; and

Figure 9 is a block diagram of a scalable, reliable SIP signaling router according to an alternate embodiment of the present invention.

P05850 09/896,559

Detailed Description of the Invention

Figure 2 is block diagram of a scalable, reliable SIP signaling router according to an embodiment of the present invention. In Figure 2, SIP signaling router **200** includes a plurality of cluster nodes **202** that perform SIP protocol functions. For example, cluster nodes **202** may comprise SIP proxy servers, SIP redirect servers, or combination proxy/redirect servers. An active location server **204** maintains a database of SIP location information and replicates the database to SIP cluster nodes **202** and to a standby location server **206**. Standby location server **206** provides a redundant copy of the SIP location database maintained by active location server **204** in the event of failure of active location server **204**. Management node **208** performs network management functions and other services, such as domain name system (DNS) service, dynamic host configuration protocol (DHCP) service, and trivial file transfer

protocol (TFTP) service. An exemplary hardware platform suitable for nodes **202**, **204**, **206**, and **208** is the NETRA™ T1 available from SUN Microsystems.

In order to provide connectivity to external networks, SIP signaling router **200** includes active Ethernet switch **210** and standby Ethernet switch **212**. Ethernet switches **210** and **212** may be connected to each other by a high-speed link **213**. High-speed link **213** may be any type of high-speed link, such as a gigabit Ethernet link. High-speed link **213** may be used for inter-switch communication, such as exchange of a connection tuple table, which will be described below. In order to provide redundant network layer connectivity to external networks, Ethernet switches **210** and **212** are connected to primary and backup IP routers **214** and **216**. In the illustrated example, each of the cluster nodes **202**, location servers **204** and **206**, and management node **208** include two Ethernet interfaces – one connected to active Ethernet switch **210** and the other connected to standby Ethernet switch **212**.

In order to provide reliability among cluster nodes, multiple cluster nodes provide redundancy for each other. In this configuration, if a cluster node fails, one or more of the other redundant load-sharing nodes will continue providing SIP service provided by signaling router **200**.

Real Time Replication of SIP Location Database

As stated above, an important feature of the invention is the fact that active location server **204** replicates its database of SIP location information to cluster nodes **202** in real time. As a result of this real time replication of the SIP location database, cluster nodes **202** can route SIP signaling messages based on their own local copies of the SIP location database. This greatly increases routing speed over conventional SIP proxy servers that depend on a centralized

Attorney Docket No. 2003-0076

location database.

Figure 3 is a block diagram of active location server **204** illustrating the steps for replicating the SIP location database to cluster nodes **202** in real time. In the example illustrated in Figure 3, active location server **204** includes a database module **300** for maintaining a SIP location database **302** and a provisioning log **304** for the SIP location database. A database provisioning module **306** provisions new SIP location information in SIP location database **302**. Provisioning module **306** may also interface with an external user to allow the user to manually input data to be provisioned in database **302**. Network provisioning module **308** replicates the SIP location database to cluster nodes **202** and standby location server **206**. Finally, maintenance module **310** controls the overall operations of active location server **204**. It is understood that modules **300**, **304**, **306**, **308**, and **310** may be implemented in hardware, software, or a combination of hardware and software.

Referring to the message flow illustrated in Figure 3, in step 1, provisioning module **306** and database module **300** communicate to update one or more records in SIP location database **302**. When the records are updated, database module **300** stores the updated records in provisioning log **304**. In step 2, database module **300** notifies network provisioning module **308** of the existence of changed database records by indicating the latest database level. In step 3, network provisioning module **308** requests real time database file records affected by the update indicated in provisioning log **304**. This step may be performed periodically.

In step 4, database module **300** sends the updated records stored in provisioning log **304** to network provisioning module **308**. In step 5, network

provisioning module **308** multicasts the database update to cluster nodes **202** and to standby location server **206**. In a preferred embodiment of the invention, the multicasting is performed via the reliable multicast protocol (RMTP) II protocol. The RMTP II protocol is described in *Reliable Multicast Transport Protocol (RMTP)*, S. Paul et al., IEEE Journal on Selected Areas in Communications, volume 15, number 3, April 1997, pages 407-421, and *RMTP: A Reliable Multicast Transport Protocol*, Lynn et al., Proceedings of IEEE INFOCOM '96, pages 1414-1424, the disclosures of each of which are incorporated herein by reference in their entirety. In addition, exemplary software for RMTP can be downloaded free of charge from www.bell-labs.com/project/rmtp/rmtp.html.

RMTP is a reliable multicast transport protocol for the Internet. RMTP provides sequenced, lossless delivery of a data stream from one sender to a group of receivers. RMTP is based on a multi-level hierarchical approach, in which the receivers are grouped into a hierarchy of local regions, with a designated receiver in each local region.

Receivers in each local region periodically send acknowledgements to their corresponding designated receiver. The designated receivers send acknowledgements to the higher-level designated receivers, until the designated receivers in the highest level send acknowledgements to the sender, thereby avoiding the acknowledgement implosion problem. Designated receivers cache receive data and respond to retransmission requests of the receivers in their corresponding local regions, thereby decreasing end-to-end latency. RMTP uses a packet-based selective repeat retransmission scheme for higher throughput.

Reliability in RMTP is achieved through a multi-level hierarchical approach in which leaf receivers periodically send status message to designated receivers. Status messages consist of the lower end of the flow control window and a bit vector indicating which packets are received and lost relative to the window's lower end. Designated receivers, in turn, send their status periodically to higher layer designated receivers and so on until the designated receivers at the highest level send their status to the sender. Lost packets are recovered by local retransmissions by their designated receiver. Retransmissions are either unicast or multicast based a threshold.

Flow control in RMTP is achieved by a combination of rate control and window based control. The sender can set its maximum rate before a session begins and then it can adjust its rate based on the status of receivers. RMTP used a TCP-like slow start mechanism when congestion is sensed (e.g., multiplicative back off and linear increase of window size).

Figure 4 is a flow chart illustrating exemplary steps performed by standby location server **206** in response to receiving an update from active location database **204**. The components of standby location server **206** are the same as those of active location server **204**. Hence, a detailed description thereof will not be repeated herein. Referring to Figure 4, in step **ST1**, standby location server **206** receives a database update from active location server **204**. In step **ST2**, standby location server **206** checks the status of its local SIP location database. In step **ST3**, if standby location server **206** determines that the database is not coherent, standby location server **206** continues to check the database status until the database is coherent. If the database is determined to be coherent, control proceeds to step **ST4** where active location server **204** validates the

database level and birth date in the received database update against the current database level. In step **ST5**, if the database level of the SIP location database is determined to be current, the update procedure ends.

In step **ST6**, if the database level is determined not to be current, standby location server **206** begins the RMTP update transaction. In step **ST7**, standby location server **206** copies the RMTP update records to its provisioning log, reads the records written into the provisioning log, and verifies that the records were correctly written with a checksum. In step **ST8**, RMTP end transaction processing begins. In step **ST9**, the SIP location database maintained by standby location server **206** is set to incoherent. The purpose of setting the database to end coherent is to prevent modification or reading by another process while the database is being updated. In step **ST10**, standby location server **206** transfers the received updates into its SIP location database. In step **ST11**, standby location server **206** commits the updates to its provisioning log. In step **ST12**, standby location server **206** sends a message to its standby network provisioning module indicating the latest database level.

Figure 5 is a flow chart illustrating exemplary steps performed by standby location server **206** in performing incremental database loading of its SIP location database. Incremental database loading occurs when standby location server **206** has missed transactions that are still contained in provisioning log **304** of active location server **204**. Referring to Figure 5, in step **ST1**, standby location server **206** receives an RMTP update message. In step **ST2**, standby location server **206** checks the database level in the update message. In step **ST3**, if standby location server **206** determines that the level is not greater than a predetermined threshold level, in step **ST4**, standby location server **206** performs

the database update procedure illustrated in Figure 4.

If standby location server **206** determines that the level in the update message is greater than the predetermined threshold, control proceeds to step **ST5**, where standby location server **206** sends an incremental load request message to maintenance module **310** on active location server **204**. Step **ST6**, maintenance module **310** on active location server **204** starts an incremental loading stream to standby location server **206** at the requested database level. In step **ST7**, network provisioning module **308** on active location server **204** protects the required records in the provisioning log, i.e., those that are the subject of the incremental loading by registering the incremental loading stream with database module **300**. Thus, the steps illustrated in Figure 5 describe incremental loading of the SIP location database on standby location server **206**.

According to another aspect of the invention, cluster nodes **202** may request continuous reloading from active location server **204**. Figure 6 illustrates exemplary steps performed by cluster nodes **202** and active location server **204** in continuously reloading the SIP location databases maintained by cluster nodes **202**. Referring to Figure 6, in step **ST1**, network provisioning module **308** on active location server **204** receives a reload request from a cluster node. In step **ST2**, database module **300** reads records from SIP location database **302** and in step **ST3**, the records are forwarded to the requesting cluster node. In step **ST4**, active location server **204** determines whether all requests have been processed. If all requests have not been processed, control returns to step **ST2** where records are read from the database and forwarded to the requesting cluster nodes. If all reload request have been processed, the reload procedure ends.

In reading records from SIP location database **302**, if active SIP location server **204** receives a request from another cluster node for reloading, active location server **304** uses the existing record stream and notifies location server provisioning module **310** of the first record read for that cluster node. This process continues until the SIP location databases on all requesting cluster nodes have been reloaded.

Another function performed by SIP signaling router according to an embodiment of the present invention is cluster node incremental loading. As stated above, incremental loading may occur when a cluster node detects that an update received from active location server **204** is greater than it is expected. Figure 7 illustrates exemplary steps performed by a cluster node and by the active location server in performing cluster node incremental loading. Referring to Figure 7, in step **ST1**, a cluster node sends a request to the active location server for incremental loading. In step **ST2**, the network provisioning module in the active location server receives the request and requests records from the location database associated with a next incremental database level above the current database level in the request. In step **ST3**, active location server **204** stores the new levels in the maintenance module and forwards the database records to the cluster nodes. In step **ST4**, the active location server determines whether the cluster node database is current. If the database is not current, in step **ST5**, active location server **206** gets the next level and steps **ST2** through **ST5** are repeated until the cluster node database is current.

According to another aspect, the present invention includes messaging systems for monitoring the operational status of cluster nodes providing SIP proxy services, load sharing between the cluster nodes, and switching between

cluster nodes in the event of a failure.

Figure 8 is a partial block/partial flow diagram illustrating a method for monitoring the operational status of cluster nodes performing SIP proxy services according to an embodiment of the present invention. In Figure 8, cluster nodes **202** are each connected to active and standby Ethernet switches **210** and **212**. Both active Ethernet switch **210** and standby Ethernet switch **212** maintain a connection tuple table **1000** that contains the following information for each connection maintained by cluster nodes **202**:

> Destination IP Address, Originating IP Address, Destination Port Number, Originating Port Number, and MAC Address of the Cluster Node Associated With the Connection.

The connection tuple tables **1000** allow active and standby Ethernet switches **210** and **212** to keep track of the number of connections maintained by each cluster node **202**.

In order to determine the operational status of cluster nodes **202**, in the illustrated embodiment, active Ethernet switch **210** sends health check and packet Internet groper (PING) message to each of cluster nodes **202**. The PING and health check messages may be sent periodically. The PING messages determine the functionality of protocol layers 1-3 of the protocol stack executing on each cluster node **202**. The health check messages determine the application level health of cluster nodes **202**. Accordingly, if a cluster node fails to respond to a PING message there is no need to send a health check message to that cluster node.

ATTORNEY DOCKET NO. 2005-0560

In addition to being useful for monitoring the operational status of the cluster node, the PING and health check messages may be used along with the connection tuple tables to perform load sharing among cluster nodes. For example, active Ethernet switch **210** may monitor the response time of each cluster node **202** for responding to a PING or health check message. The response time is indicative of the load on each cluster node **202**. The connection tuple table **1000** could be used to determine the number of connections maintained by each cluster node. Load sharing may be performed based on the response time and the number of connections managed by a given cluster node. For example, it may be desirable to increase message flow to a cluster node that responds quickly and has a small number of connections in its connection table. Any combination of response time and number of active connections may be used as a basis for load sharing.

When one of the Ethernet switches **210** and **212** fails or when one or more ports associated with switches **210** and **212** fail, it may be desirable to switch to the other Ethernet switch or port. According to the present invention, Ethernet switches **210** and **212** include a trunking capability that allows switch over from one Ethernet switch to the other Ethernet switch in the event of failure. This trunking capability is described in IEEE 802.3ad, the disclosure of which is incorporated herein by reference in its entirety.

IEEE 802.3ad includes a link aggregation standard that provides inherent, automatic redundancy on point-to-point links. In other words, should one of the multiple ports used in a link fail, network traffic is dynamically redirected to flow across the remaining good ports in the link. The redirection is fast and triggered when a switch learns that a media access control address has been

automatically reassigned from one link port to another in the same link. The switch then sends the data to the new port location, and the network continues to operate with virtually no interruption in service.

The emerging IEEE 802.3ad specification will deliver switch-to-switch and switch-to-server incremental bandwidth increases in a way that also brings inherent failover capabilities to Ethernet networks. Link aggregation works by making two to six or more physical links appear as a single logical link to Spanning Tree and any other Layer 2 or 3 protocol. At the same time, link aggregation makes automatic failover possible by enabling the physical links to serve as redundant backups to one another.

The 802.3ad specification adds a link aggregation sublayer to the conventional Ethernet protocol stack at Open Systems Interconnection Layer 2, the media access control (MAC) layer. This sublayer effectively separates the physical connections below from the new, logical MAC address it shows to higher level protocols. Within the sublayer, a link aggregation control protocol (LACP) performs functions that range from verifying configurations and operating status of participating devices to carrying out the distribution tasks necessary for assigning packet flows to their physical links.

The LACP also carries out the collection tasks necessary for receiving incoming packets. Also, the protocol contains a control function for adding and deleting physical links. The distribution mechanism determines which packet flows will go over which physical links. In the event of a link failure, the control function alerts the distributor, which then reassigns the packet flows. Because the operations are carried out low in the OSI protocol model, failure detection and reselection can occur very quickly, typically in less than a second.

Switches **210** and **212** of SIP signaling router **200** illustrated in Figure 2 may utilize the IEEE 802ab link aggregation control protocol to dynamically re-route SIP signaling traffic around congested a failed links. For example, switch **210** may have multiple physical links connected to one of the cluster nodes that are aggregated into a single logical link using the LACP protocol. When switch **210** detects or is notified of a failure of one of the physical links, traffic is dynamically redirected to another physical link within the logical link. This dynamic redirection is accomplished at the link aggregation sublayer, and as a result, is transparent to higher layers.

Figure 9 illustrates an alternate embodiment of a SIP signaling router according to the present invention. In the illustrated embodiment, SIP signaling router **200A** comprises a plurality of printed circuit boards connected via a communications bus. Each printed circuit board includes one or more microprocessors. For example, each printed circuit board may include an application processor for performing SIP functions and a communications processor for communicating via the communications bus. In the illustrated example, active location server **204A** replicates its local database of SIP location information to cluster nodes **202A** via the communications bus. In a preferred embodiment of the invention, communications bus **1100** comprises a dual, counter rotating serial bus. Local subsystem management system (LSMS) **1102** provisions the data stored in the SIP location database managed by active location server **204A**. LSMS **1102** may also interface with an external device to receive database information from a user.

The underlying hardware illustrated in Figure 9 is similar to the hardware architecture of an EAGLE® signal transfer point available from Tekelec of

Calabasas, California. However, rather than performing signaling system seven routing functions, the SIP signaling router illustrated in Figure 9 performs SIP routing functions.

It will be understood that various details of the invention may be changed without departing from the scope of the invention. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation—the invention being defined by the claims.

EXPRESS MAIL NO. EV 339 526 950